

PK4



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/620,714	07/20/2000	Luke Matthew Browning	AUS9-2000-0277-US1	3354

7590 07/30/2003

Duke W yee
Carstens Yee & Cahoon LLP
P O Box 802334
Dallas, TX 75380

EXAMINER

STEELMAN, MARY J

ART UNIT	PAPER NUMBER
----------	--------------

2122

DATE MAILED: 07/30/2003

6

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary

Application No.

09/620,714

Applicant(s)

BROWNING ET AL.

Examiner

Mary J. Steelman

Art Unit

2122

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --
Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133).
- Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☒ Responsive to communication(s) filed on 02 June 2003.
- 2a) ☒ This action is **FINAL**. 2b) ☐ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) ☒ Claim(s) 1-29 is/are pending in the application.
- 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) ☐ Claim(s) _____ is/are allowed.
- 6) ☒ Claim(s) 1-29 is/are rejected.
- 7) ☐ Claim(s) _____ is/are objected to.
- 8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on _____ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.
- Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
- 11) ☐ The proposed drawing correction filed on _____ is: a) ☐ approved b) ☐ disapproved by the Examiner.
- If approved, corrected drawings are required in reply to this Office action.
- 12) ☐ The oath or declaration is objected to by the Examiner.

Priority under 35 U.S.C. §§ 119 and 120

- 13) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some * c) ☐ None of:
- ☐ Certified copies of the priority documents have been received.
 - ☐ Certified copies of the priority documents have been received in Application No. _____.
 - ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).
- * See the attached detailed Office action for a list of the certified copies not received.
- 14) ☐ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. § 119(e) (to a provisional application).
- a) ☐ The translation of the foreign language provisional application has been received.
- 15) ☐ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. §§ 120 and/or 121.

Attachment(s)

- 1) ☐ Notice of References Cited (PTO-892)
- 2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
- 3) ☐ Information Disclosure Statement(s) (PTO-1449) Paper No(s) _____
- 4) ☐ Interview Summary (PTO-413) Paper No(s). _____
- 5) ☐ Notice of Informal Patent Application (PTO-152)
- 6) ☐ Other: _____

DETAILED ACTION

1. This action is in response to the amendment filed 06/02/03.
2. As per Applicant's request, claims 1, 6, 13, 18 and 25 have been amended. New claims 26-29 have been added. Claims 1-29 are pending.

Claim Rejections - 35 USC § 102

3. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

4. **Claims 1-6, 11-18, & 23-29** are rejected under 35 U.S.C. 102(e) as being anticipated by U.S. Patent 6,240,529 to Kato.

Per claims 1, 6, 13, 18 & 25:

Kato disclosed a method, apparatus (col. 6, line 11) and a program recorded on a recording medium (col. 7, line 5) for debugging a process and storing state information.

-initiating debugging of the process; ((Col. 7, line 15, "When debugging is started...");

Art Unit: 2122

-saving a process state in response to a first event to form a stored process state; (Col. 7, lines 65-66, “storing a debugged state into a file if a certain event occurs.”);

-retrieving the stored process state in response to a predefined event; and reinitiating debugging from the stored process state. (Col. 8, lines 29-35, “When a state restoration command is issued, storage situation management unit displays a list of currently existing state storage files based on storage situation management file. Fig. 4 shows an example of the list. If a file to be restored is selected from the information, then state restoration unit reads in the file (retrieves) and restores (reinitiates) the debugged state.” Fig. 3, Execution units #108 & #109 (processor) and storage file, #118 (memory). Fig. 4, Execution upon restoration. Figure 5B, #333.)

Per claims 2 and 14:

-first event occurs periodically. (Col. 6, lines 14-19, “storage...for...debugged state...can be designated at an arbitrary point of time...”)

Per claims 3 and 15:

-process state is saved in a checkpoint data structure. (Col. 10, lines 30-32, “debugged state is stored, a state storage file...”)

Per claims 4 and 16:

-checkpoint data structure is a checkpoint file. (Col. 10, lines 30-32, “debugged state is stored, a state storage file...”)

Per claims 5 and 17:

-checkpoint data structure includes a process descriptor for the process. (Figures 4 & 6, State Storage File Name, and col. 5, lines 43-45, “storing a state storage file name and the situation upon the storage in a correlated condition into a storage situation management file...”)

Art Unit: 2122

Per claims 11 and 23:

-the process state is saved when the program is in a stopped state. (Col. 7, lines 65-66, “storing a debugged state into a file if a certain event occurs.”)

Per claims 12 and 24:

-the stopped state is at a breakpoint. (Col. 7, lines 37-38, “A break point at which execution of a program is interrupted...” Also, col. 10, lines 20-26.)

Per claim 26:

-the first event is a breakpoint and the predefined event is a checkpoint, and further comprising the step of repeatedly running between the checkpoint (event registered) and the breakpoint for a plurality of times. (Fig. 5A, #309 and #314, and Col. 9, lines 15-36, “Then, it is checked whether or not a state corresponding to one of events registered (checkpoint) by the user is satisfied...If a request for a break (breakpoint) is generated in response to the detection of the event...”)

Per claim 27:

-variable values are automatically modified after retrieving the stored process state for the checkpoint. (Col. 8, lines 33-35, “If a file to be restored is selected from the information, then state restoration unit reads in the file (modify values) and restores the debugged state.” Also see fig. 3, #117, state restoration unit.)

Per claim 28:

-initiating a debug process; (Col. 7, lines 15.)

Art Unit: 2122

-creating a child process from the debug process; (Col. 10, lines 43-46, "...the storage situation management unit...additionally has a function of recording a debugged state storage timing into trace information.")

-saving a process state of the child process in response to a first event, to form a stored process state; (Col. 10, lines 49-51, "...information of the state storage file name and so forth is stored also into a buffer (saving state of child process) for trace information storage." Also see fig. 6)

-retrieving the stored process state in response to a second event; executing the child process using the stored process state (See fig. 5B, #333 & 'A' for re-executing). (Col. 11, lines 19-29, "...the debugged state storage file name is recorded into 407 (fig. 6) at a timing at which the debugged state is stored...the retrieval means which refers to trace information is additionally provided, and specification of a debugged information file to be restored is facilitated by confirming a state storage file, a type of an execution instruction upon production of the file, a state of memory and so forth.")

Per claim 29:

-tracing a process by a debugger; (Col. 7, lines 35-37, "Debugger unit in debugging apparatus performs processing regarding a debugging function for breaking, tracing or the like.")

-saving a process state of the traced process and a process state of another process that is not being traced by the debugger; (Col. 7, lines 35-47 and 65-66, "...an instruction execution situation, a memory access situation and so forth at the point of time are recorded...")

-retrieving the saved process states; (Col. 8, lines 29-35, "When a restoration command is issued...")

Art Unit: 2122

-reinitiating debugging of the process using the retrieved process states. (Col. 8, lines 29-35, “When a state restoration command is issued, storage situation management unit displays a list of currently existing state storage files based on storage situation management file. Fig. 4 shows an example of the list. If a file to be restored is selected from the information, then state restoration unit reads in the file and restores the debugged state.” Also col. 10, lines 33-36, “...an example wherein more precise retrieval of a debugged state to be restored is allowed by combining them with another debugging function.” Also, figures 5A and 5B, reinitiate debugging @ ‘A’)

Claim Rejections - 35 USC § 103

5. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

6. **Claims 7-9 and 19-21** are rejected under 35 U.S.C. 103(a) as being unpatentable over U.S. Patent 6,240,529 to Kato, and further in view of U.S. Patent 5,560,009 to Lenkov et al. Kato disclosed an invention to debug and save state to a file, allowing resumption. Kato failed to provide information on data type descriptors, instance descriptors, and data block corresponding to an instance descriptor.

However, Lenkov described an invention that generated symbolic debug information, including creating a debug data structure (col. 6, line 62), and (col. 7, lines 33-48) including various information...debug name and type tables, a table of name strings, and object file symbol

table. Col. 8, lines 39-46, “The core object file class is used to read common object file data structures...includes access routines to look up symbols in symbol tables... (Col. 16, lines 57-67), “the dtab class includes member for holding state during iterations...The classes, data structures, variables and functions which comprise the dtab... (Col. 17, lines 55-60) blocktab is a derived class of dtab. Blocktab is the base class of all source blocks... (Col. 25, lines 46-51) The output object file interface creates a new object/debug file and stores...debug information.” Also, (Col. 26, lines 20-21) “output object file interface is implemented in the form of classes...instantiations...” Also, (Col. 26, lines 56-58) “the preprocessor creates a source file descriptor table, a procedure descriptor table, a class descriptor table...”

Therefore, it would have been obvious, to one of ordinary skill in the art, at the time of the invention, to modify Kato’s invention to debug, store state information, and restore state, by including Lenkov’s invention that processed debug information, because it arranges symbolic debug information for storage and retrieval in a meaningful manner for efficient diagnostics..

7. **Claims 10 and 22** are rejected under 35 U.S.C. 103(a) as being unpatentable over U.S. Patent 6,240,529 to Kato, and further in view of U.S. Patent 6,412,106 to Leask et al..

Kato disclosed an invention to debug and save state to a file, allowing resumption. Kato failed to provide information regarding modifications prior to resuming a debug process.

However Leask disclosed a debugging invention that allows for modifying values while the program is suspended.

Therefore, it would have been obvious, to one of ordinary skill in the art, at the time of the invention, to modify Kato’s invention to debug and store state, by including the features in

Art Unit: 2122

Leask's invention that allow for modifying values because it would create a quicker interactive debug session, saving time by not requiring a start from the beginning of a program.

Per claims 10 and 22:

-modifying at least one register or memory variable before resuming debugging from the stored process state. (Leask, col. 12, lines 55-56, "capability to modify values stored in variables, while the application program is suspended.")

Response to Arguments

8. Applicant's arguments filed 06/02/2003 have been full considered but they are not persuasive.

Applicants have argued, in substance, the following:

(A) Regarding claims 6, 18, 28, and 29: The claimed invention is a debugging method and system for a plurality of processes, whereas Kato reference is directed to debugging a single program. This is not a system having a plurality of processes running concurrently.

Examiner directs Applicant to the first sentence of the Specification. "The present invention relates to an improved data processing system and, in particular, to a method and apparatus for debugging a program or programs." Furthermore, it is well known that when a program is executing, multiple processes are executing in the background concurrently.

(B) Regarding claims 1, 13, and 25: These claims refer to abilities to retrieve a process state in response to a predefined event, allowing automation of a checkpoint/restore sequence with automatic modification of variables. The claimed feature allows for a process to automatically run repeatedly between a checkpoint and breakpoint using a plurality of register or memory variable values, thereby improving debugging capabilities. The Kato reference teaches that state

Art Unit: 2122

information can be stored, but requires a manual restoration of a state storage file, responsive to user input. Applicant argues that this is not “automatic retrieval of state information.”

Examiner responds that the Kato reference shows at ‘A’ in figures 5A and 5B where a process automatically runs repeatedly between a checkpoint and breakpoint using a plurality of register or memory variable values. Examiner points to col. 8, lines 29-35, “When a state restoration command is issued...” Kato is not specific as to how the command is issued. Furthermore, “automatic retrieval of state information” is not a claim limitation. Additionally, no matter whether a user inputs a command to restore, or the system inputs a command to restore, there is some element of “automatic” involved in reading the state from the file back into the executing program.

(C) Regarding claims 6 and 18: The limitations “wherein the process has control over at least one child process and the process state includes a process descriptor for each of the at least one child process” are not found in the Kato reference.

Examiner disagrees. A child process can be found at col. 10, lines 33-41. “...also setting such that trace information is acquired upon detection of the same event is possible.” In Embodiment 2, additional trace information is stored into a buffer.

Therefore the rejection of claims 1-29 is proper and maintained herein.

Conclusion

9. THIS ACTION IS MADE FINAL. Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

Art Unit: 2122

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.


10. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Mary Steelman, whose telephone number is (703) 305-4564. The examiner can normally be reached Monday through Thursday, from 7:00 A.M. to 5:30 P.M. If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Tuan Dam can be reached on (703) 305-4552.

The fax phone numbers are (703) 746-7240 for regular communications and (703) 746-7239 for After Final communications. Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the receptionist whose telephone number is (703) 305-3900.

Mary Steelman



07/24/2003


TUAN Q. DAM
PRIMARY EXAMINER